PATENT APPLICATION
Docket No. MS1-577US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of		)
		)
	Bailey et al.	)
		)
Serial No.:	09/638,658	) Appeal No.
		)
Confirmation No.	9652	)
		)
Filed:	August 14, 2000	)
		)
For:	Methods and Arrangements for Selective Placement of Movable Objects within a Graphical User Interface	)
		)
Examiner:	Peng Ke	)

The Honorable Commissioner of Patents
Mail Stop Appeal Brief - Patents
P.O. BOX 1450
Alexandria, VA 22313-1450

BRIEF OF APPELLANT

The Applicant has filed a timely Notice of Appeal from the action of the Examiner in finally rejecting all of the claims that were considered in this application. This Brief is being filed under the provisions of 37 C.F.R. § 1.192. The Filing Fee, as set forth in 37 C.F.R. § 1.17(c), is submitted herewith.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation, by way of assignment from Bailey et al, who is the named inventive entity and is captioned in the present brief.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF CLAIMS

Claims 1-27 are pending and are the subject of this appeal.

<u>STATUS OF AMENDMENTS</u>

None.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Graphical user interfaces (GUIs) are popular in a variety of computing applications and/or operating systems. In certain instances, physical user interfaces, such as, e.g., knobs, buttons, switches, handles, and the like, can be graphically modeled and included within the graphical user interface as selectable/movable objects. One type of physical input device is a touch pad or touch screen. Touch screen are usually configured to allow the user to touch an exposed surface of a display device, through which the GUI environment and GUI objects are visible. Unfortunately, when a GUI object is selected in a conventional touch screen GUI environment, the position of the pointing device, e.g., stylus, fingertip, etc. may not always be at or near the center or other preferred "grab point" of the selected object. *See Application, Page 1, Line 6 to Page 2, Line 13.*

For example, a user may select a slider portion of a modeled sliding control knob at a corner or along an edge. Conventional GUI environments model this unnatural capability, while others try to automatically correct the situation. Thus, for example, in some cases, an offset between the user input position and the GUI object is determined and then maintained throughout the movement/activation process. In other cases, the GUI object (e.g., the slider knob) is immediately relocated within the GUI environment in such a manner that it is "correctly" positioned in accord with the user's input. While each of these implementations tends to work just fine, many users may find the maintained offset or the immediate correction less pleasing. *See Application, Page 2, Line 14 to Page 3, Line 7.*

To address these problems and to provide a more visually appealing GUI experience, methods and arrangements are provided below that allow for a gradual visual and operative correction between the positioning information and GUI object 110. For example, if an offset exists between a user's input and an object's preferred contact area 266, then, as the user moves GUI object 110 within GUI environment 108 the magnitude of the offset is selectively and incrementally altered over time/distance in a manner that is nearly imperceptible to the user. This corrective process is continued, until the latest positioning information significantly matches preferred contact area 266 or falls within a defined tolerance thereof, or the user stops providing new positioning information. Consequently, the correction process tends to be more pleasing and in some instances more realistic than an immediate correction or maintenance of a continued offset. *See Application, Page 9, Lines 9-21.*

Independent Claim 1 recites a method comprising in a graphical user interface (e.g., reference number 108, FIGS. 1 and 3; page 5, lines 6-17):

- determining an offset value between a selected object's position and an input position (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c), reference number 306, FIG. 5; Page 9, Line 13 to page 10, line 11, Page 11, Lines 7-14); and
- dynamically and gradually reducing the offset value by correctively adjusting the input position with respect to the object's position in proportion to a movement of

the selected object (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c); reference number 308, FIG. 5; page 10, line 12 to page 11, line 3, Page 11, Lines 7-14).

Independent Claim 8 recites a method comprising in a graphical user interface (e.g., reference number 108, FIGS. 1 and 3; page 5, lines 6-17):

- determining an offset value between a selected object's position and an input position, wherein the input position includes updated positioning information from a user input mechanism and wherein the selected object moves in proportion to a change in the positioning information(e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c), reference number 306, FIG. 5; Page 9, Line 13 to page 10, line 11, Page 11, Lines 7-14); and
- dynamically and gradually reducing the offset value by implementing a corrective function including a linear corrective factor that selectively and incrementally reduces the offset in proportion to a movement of the selected object (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c); reference number 308, FIG. 5; page 10, line 12 to page 11, line 3, Page 11, Lines 7-14).

Independent Claim 10 recites a computer-readable medium (e.g., reference number 222, FIG. 2; page 6, lines 1-23) having computer-executable instructions for causing at least one processing unit to perform acts comprising:

- determining an offset value between a selected object's position and an input position (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c), reference number 306, FIG. 5; Page 9, Line 13 to page 10, line 11, Page 11, Lines 7-14); and
- in proportion to a movement of the selected object, dynamically and gradually reducing the offset value by correctively adjusting the input position with respect to the object's position (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c); reference number 308, FIG. 5; page 10, line 12 to page 11, line 3, Page 11, Lines 7-14).

Independent Claim 16 recites a computer-readable medium (e.g., reference number 222, FIG. 2; page 6, lines 1-23) having computer-executable instructions for causing at least one processing unit to perform acts comprising:

- determining an offset value between a selected object's position and an input position(e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c), reference number 306, FIG. 5; Page 9, Line 13 to page 10, line 11, Page 11, Lines 7-14); and
- dynamically and gradually reducing the offset value using a corrective function that selectively and incrementally reduces the offset in proportion to a movement

of the selected object (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c); reference number 308, FIG. 5; page 10, line 12 to page 11, line 3, Page 11, Lines 7-14).

Independent Claim 17 recites an apparatus (e.g., reference number 200, FIG. 2) comprising logic (e.g., reference number 235-238, FIG. 2; page 6, lines 20-23) configured to determine an offset value between a selected object's position and an input position (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c), reference number 306, FIG. 5; Page 9, Line 13 to page 10, line 11, Page 11, Lines 7-14), and dynamically and gradually reduce the offset value by correctively adjusting the input position with respect to the object's position in proportion to a movement of the selected object (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c); reference number 308, FIG. 5; page 10, line 12 to page 11, line 3, Page 11, Lines 7-14).

Independent Claim 23 recites an apparatus (e.g., reference number 200, FIG. 2) comprising:

- a display device having a plurality of pixels (e.g., reference number 247, FIG. 2; page 7, lines 8-11);
- an input device configured to generate updated positioning information within an input position (e.g., reference numbers 255, 257, FIG. 2; page 6, line 24 to page 7,

line 11);

- logic (e.g., reference number 235-238, FIG. 2; page 6, lines 20-23) operatively coupled to the display device and the input device and configured to determine an offset value between a selected object's position and the input position (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c), reference number 306, FIG. 5; Page 9, Line 13 to page 10, line 11, Page 11, Lines 7-14), and reduce the offset value using a corrective function that selectively and incrementally reduces the offset in proportion to a movement of the selected object based on the updated positioning information (e.g., reference numbers 260, 266 and 270, FIGS. 4(1)-4(c); reference number 308, FIG. 5; page 10, line 12 to page 11, line 3, Page 11, Lines 7-14).

<u>GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL</u>

1. Claims 1-25 and 27 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,808,601 to Leah et al. (hereinafter "Leah") in view of U.S. Patent No. 6,894,678 to Rosenberg et al (hereinafter "Rosenberg").

2. Claim 26 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Leah in view of Rosenberg in view of U.S. Patent No. 5,870,083 to Shieh et al (hereinafter "Shieh").

ARGUMENT

First Ground of Rejection Claims 1-25 and 27 satisfy the requirements of 35 U.S.C. § 103(a) and therefore are patentable over Leah in view of Rosenberg.

Beginning at page 2 of the subject Applicant, problems with tradition techniques for interaction with a GUI are described. When a GUI object is selected in a conventional touch screen GUI environment, the position of the pointing device, e.g., stylus, fingertip, etc. may not always be at or near the center or other preferred "grab point" of the selected object. For example, a user may select a slider portion of a modeled sliding control know at a corner or along an edge. Nevertheless, many conventional GUI environments model this unnatural capability, while others try to automatically correct the situation. Thus, for example, in some cases, an offset between the user input position and the GUI object is determined and then maintained throughout the movement/activation process. In other cases, the GUI object (e.g., the slider know) is immediately relocated within the GUI environment in such a manner that it is "correctly" positioned in accord with the user's input.

Methods and arrangements are claimed that allow for a gradual visual and operative correction between the positioning information and GUI object. For example, if an offset exists between a user's input and an object's preferred contact area, then, as the user moves GUI object within GUI environment the magnitude of

the offset is selectively and incrementally altered over time/distance in a manner that is nearly imperceptible to the user. This corrective process is continued, until the latest positioning information significantly matches preferred contact area or falls within a defined tolerance thereof, or the user stops providing new positioning information. Consequently, the correction process tends to be more pleasing and in some instances more realistic than an immediate correction or maintenance of a continued offset. *See Application, Page 9, Lines 9-21.*

The Examiner first asserts Leah for "determining an offset value between a selected object's position and an input position; and dynamically and gradually reducing the offset value by correctively adjusting the input positioned with respect to the object's positions (fig 1, 2c, col. 6, lines 16-44)". *See Office Action Dated April 4, 2006, Page 2.* The Applicant respectfully disagrees. Leah, however, discloses a technique for *selecting* a *non-moving* visual object on a graphical user interface (GUI) (i.e., a screen). Leah does not disclose a technique that can be used with already *selected* objects. Likewise, Leah does not disclose a method that depends on *movement* of the object.

Specifically, Leah discloses a technique for widening the boundary around a visual object on a screen for purposes of making selection of the object easier—since the "selectable" or "clickable" object becomes bigger than the visual object. This greater footprint around an object on a GUI makes the object an easier target for

selection by a pointing arrow of a mouse than an unadulterated object. Further, once a selection agent, such as the pointing arrow of a mouse, is inside the expanded boundary, the Leah technique instantly gravitates the selection agent (such as the mouse pointing arrow) to the center of the visual object. Hence, the Leah technique aims to make it easier to select an object.

Claim 1, on the other hand, in a first distinction from Leah, recites that an object that has already been *selected* (e.g., clicked on once by a mouse; or selected by touching the object, in a touch pad system). In a further distinction from Leah, the method of Applicant's claim 1 performs when the object is *moving*. It is respectfully submitted that the Examiner has not taught or suggested where these claimed features are met in Leah nor any of the other submitted references, and therefore a *prima facie* case of obviousness has not been established.

Nevertheless, the Examiner correctly admits that "Leah et al. fails to teach in proportion to a movement of the selected object". *See Office Action Dated April 4, 2006, Page 2.* To correct this defect, the Examiner asserts the following portion of Rosenberg for such teaching, which is excerpted as follows for the sake of convenience:

> FIG. 14 is a flow diagram illustrating another embodiment **650** for providing an enhanced degree of cursor control without distorting force feedback. In this embodiment, force detents are provided in a region around the cursor when the user is believed to need to finely position the cursor.

The process begins at **652**. In step **654**, the mouse position in the local frame **30** is read by the local microprocessor **130** and is the reference position. In step **656**, the process examines the previous positions of the mouse to determine the velocity of the mouse. This is similar to the procedure that the ballistics steps in FIGS. 6 and 8 and step **606** of FIG. 12 perform to determine velocity.

In step **658**, the process determines whether the mouse velocity is less than a threshold velocity, and whether the mouse has been under the threshold velocity for greater than a predetermined time period. The threshold velocity is preferably some small velocity below which the user typically desires to finely position the cursor in the graphical environment. The predetermined time period is preferably a time period found to typically pass when the user is having trouble acquiring a target or performing some other fine positioning task (and which can depend on the task). For example, a time period of 3 seconds for a particular task might be used. In an alternative embodiment, only the velocity of the mouse is checked in step **658** and the time period is ignored.

If the mouse velocity is above the threshold velocity or is not under the threshold velocity for the minimum time, the process returns to step **654** (of course, forces caused by other interactions of the cursor in the GUI or other events can be output as described above). **If the mouse velocity is less than the threshold velocity for the minimum time, then the process continues to step 660, where a field of multiple force detents are provided in a determined spacing over a determined area or region. Thus, the detents are not provided if the mouse is moving over the threshold velocity, since they would only encumber fast, coarse motion of the mouse and cursor.** However, if the user is moving the mouse slowly for the predetermined time period, the local processor assumes that the user needs assistance in fine positioning, and provided the field of force detents. The detents are preferably similar to the detents described with reference to FIG. 13, and output forces to slow quick motion of the mouse and cursor. The force detents can be provided in a rectangular grid, a series of circular radii, or in other configurations. These configurations can be predetermined, selected by the user, or may vary depending on the nearest region or object in the GUI. The field of detents can cover the

> entire screen or display frame, or may be provided only in a predefined smaller region surrounding the cursor in a predetermined shape or a shape that differs according to the region or nearest object of the GUI. In addition, large detents or small detents can be provided, and the spacing of the detents from each other can be varied as desired. For example, a grid of detents can be provided that corresponds to a grid of snap points displayed on the screen by a drawing program. In a word processor, the detents can correspond to letter spacing and line spacing of the current document. Each detent can also correspond to each pixel displayed on the screen. Ideally, the detents are spaced at the minimum resolution required for a give positioning task. For example, sensors **62** on the mouse **12** can track 1000 points per square inch. This high resolution is not required for the host computer, since, for example, 300 pixels are displayed per square inch (300 dpi). Thus, detents need only be provided at the 300 per square inch resolution. For some tasks, detent spacing greater than the pixel spacing can be provided.
>
> The local microprocessor can provide the detent field entirely independently from the host computer. Alternatively, the host computer can send high level commands to enable the force detent feature and to characterize the detent spacing, force intensity, and other parameters of the detents (thus allowing the user to enable and/or characterize detents if desired). *(emphasis added).*

Thus, as shown in the above excerpted portion, Rosenberg merely describes **the movement of a mouse** and **force feedback to the mouse** that is output **in relation to movement of the mouse**. Thus, in neither this section nor elsewhere in Rosenberg can teaching or suggestion be found for reducing an offset value **within a graphical user interface**. Rather, Rosenberg describes movement of a mouse and force feedback based on that movement outside of the graphical user interface.

It is also respectfully submitted that the Examiner has misinterpreted the features of Claim 1. For example, the motivation provided by the Examiner for the combination asserts that it "would have been obvious to an artisan at the time of the invention to include Rosenberg's teaching with method of Leah in order to reduce

user's undesired experience of any hard, physical stops when the mouse reaches a physical limit". *See Office Action Dated April 4, 2006, Page 2.* It is unclear how such motivation could be used to arrive at the recited features of Claim 1 and further supports that even if the combination could be made (which the Applicant respectfully submits that it cannot), that such a combination would not teach or suggest the claimed features.

In the *Response to Arguments* section, the Examiner asserts the following:

> In this case, Rosenberg moves a cursor against a weak magnitude resistive force when the motion of mouse is fast (column 37, line 65, column 38, lines 18) and increases the magnitude resistive force when the mouse slows. (column 37, line 65-column 38, lines 18) Therefore, by adjusting the magnitude resistive force, Rosenberg is in fact adjusting object's position in proportion to a movement of the object. Furthermore, when the mouse's speed is below a minimum threshold, Rosenberg uses multiple force detects to determine cursor's position on screen. (column 40, lines lines58-column 41 lines 32) However when the mouse's speed is above the minimum threshold, the detents were not used. (column 40, lines 58-column 41 lines 32) The use of detents also suggests adjustment of cursor's position proportionally to a movement of the mouse. *See Office Action Dated April 4, 2006, Page 8.*

Therefore, the Examiner seems to assert that cursor is the "selected object" and the "input position" is the mouse. Again, however, Claim 1 recites "a method comprising **in a graphical user interface**". Therefore, the graphical user interface is positively recited, in which, the "determining" and the "dynamically and gradually reducing" are performed. Thus, the relation between the detents and the cursor asserted by the Office is immaterial as the input position (e.g., the mouse) is not within the recited graphical user interface.

Hence, Applicant's claim 1 and the Leah technique (alone or in combination with Rosenberg) can be considered mutually exclusive on two different counts. First, the Leah/Rosenberg technique pertains to unselected objects while Applicant's method of claim 1 is directed toward selection of objects. Second, the Leah reference applies the Leah technique to unmoving objects and Rosenberg does not correct this defect, alone or in combination. Applicant's method of claim 1, on the other hand, comes into play when the object is moving (e.g., being moved). Further, the mouse is not "in a graphical user interface" as recited in claim 1. Thus, Rosenberg does not correct these defects, as Rosenberg is limited to actions performed outside of a graphical user interface and in no way discusses a relation between objects in the user interface itself.

Claims 2-7 depend either directly or indirectly from Claim 1 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in Claim 1, are neither shown nor suggested in the references of record, either singly or in combination with one another.

Independent Claims 8, 10, 16 and **23** are allowable for at least the same reasons as discussed in relation to Claim 1, and therefore the Applicant will not further burden the record. **Claims 9, 11-15, 17-22, 24, 25** and **27** depend either directly or indirectly, respectively, from Claims 8, 10, 16 and 23 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in Claims 8, 10, 16 and 23, are neither shown nor suggested in the references of record, either singly or in

combination with one another. Accordingly, the Applicant respectfully requests the Board to overturn the First Ground of Rejection.

<u>Second Ground of Rejection</u> Claim 26 satisfies the requirements of 35 U.S.C. § 103(a) and therefore are patentable over Leah in view of Rosenberg in view of Shieh.

Claim 26 depends from claim 20, which in turn depends from base claim 17. The features of claim 26, including the features of claims 20 and base claim 17, as previously described, are not taught or suggested by Leah and/or Rosenberg, alone or in combination. Shieh does not correct this defect.

Base claim 17 defines an apparatus comprising logic configured to determine an offset value between a selected object's position and an input position, and dynamically and gradually reduce the offset value by correctively adjusting the input position with respect to the object's position *<u>in proportion to a movement of the selected object</u>*. As previously described, Leah and Rosenberg do not teach or suggest claim 17's element of dynamically and gradually reducing the offset value in proportion to a movement of the selected object. In fact, Leah teaches away from this element of claim 17 because Leah teaches that if a visually displayed pointer of an input device crosses a perimeter boundary calculated at a distance around the outside of the object (col. 5, lines 37-47), then the visually displayed pointer immediately moves at once to the "hot or *<u>selectable</u>* portion of the object" (col. 5, lines 56-63) (emphases added). Shieh does not cure these defects, alone or in combination with Leah and Rosenberg.

Shieh teaches an apparatus including an input device with a touch screen, but Shieh does not teach or suggest claim 17's element of reducing the offset value in proportion to a movement of the selected object. The touch screen of Shieh does not add anything to the missing teaching (it too fails to teach or suggest the features of

claim 17), hence the combination fails to produce a prima facie obviousness rejection. Therefore, Applicant respectfully requests that the obviousness rejection be removed from claim 26 and claim 26 be allowed. Accordingly, the Applicant respectfully requests that the Board overturn the Second Ground of Rejection.

CONCLUSION

The Applicant respectfully considers this application to be in condition for allowance and respectfully requests the Board to overturn the final rejection and that the Examiner pass this application to allowance.

Dated this 23rd day of October, 2006.

Respectfully submitted,

WILLIAM J. BREEN, III
Attorney for Applicant
Registration No. 45,313

LEE & HAYES PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201
Telephone: (509) 324-9256 (Ext. 249)
Facsimile: (509) 323-8979

APPENDIX: CLAIMS ON APPEAL

1. A method comprising in a graphical user interface:

determining an offset value between a selected object's position and an input position; and

dynamically and gradually reducing the offset value by correctively adjusting the input position with respect to the object's position in proportion to a movement of the selected object.

2. The method as recited in Claim 1, wherein the object position includes a preferred contact area.

3. The method as recited in Claim 2, wherein the preferred contact area includes a definable point associated with an object, and the object can be selectively moved within the graphical user interface.

4. The method as recited in Claim 1, wherein the input position includes updated positioning information from a user input mechanism.

5. The method as recited in Claim 4, wherein dynamically and gradually reducing the offset value further includes implementing a corrective function that selectively and incrementally reduces the offset based on the updated positioning information.

6. The method as recited in Claim 4, wherein implementing the corrective function that selectively and incrementally reduces the offset based on the updated positioning information is further selectively implemented based upon differences between the updated positioning information with respect to previous positioning information.

7. The method as recited in Claim 5, wherein the corrective function includes a linear corrective factor.

8. A method comprising in a graphical user interface:

determining an offset value between a selected object's position and an input position, wherein the input position includes updated positioning information from a user input mechanism and wherein the selected object moves in proportion to a change in the positioning information; and

dynamically and gradually reducing the offset value by implementing a corrective function including a linear corrective factor that selectively and incrementally reduces the offset in proportion to a movement of the selected object.

9. The method as recited in Claim 1, further comprising graphically displaying the object within a graphical user interface.

10. A computer-readable medium having computer-executable instructions for causing at least one processing unit to perform acts comprising:

determining an offset value between a selected object's position and an input position; and

in proportion to a movement of the selected object, dynamically and gradually reducing the offset value by correctively adjusting the input position with respect to the object's position.

11. The computer-readable medium as recited in Claim 10, wherein the object position includes a preferred contact area.

12. The computer-readable medium as recited in Claim 11, wherein the preferred contact area includes a definable point associated with an object that can be selectively moved within the graphical user interface.

13. The computer-readable medium as recited in Claim 10, wherein the input position includes updated positioning information from a user input mechanism.

14. The computer-readable medium as recited in Claim 13, wherein dynamically and gradually reducing the offset value further includes implementing a corrective function that selectively and incrementally reduces the offset based on the updated positioning information.

15. The computer-readable medium as recited in Claim 14, wherein the corrective function includes a linear corrective factor.

16. A computer-readable medium having computer-executable instructions for causing at least one processing unit to perform acts comprising:

determining an offset value between a selected object's position and an input position; and

dynamically and gradually reducing the offset value using a corrective function that selectively and incrementally reduces the offset in proportion to a movement of the selected object.

17. An apparatus comprising logic configured to determine an offset value between a selected object's position and an input position, and dynamically and gradually reduce the offset value by correctively adjusting the input position with respect to the object's position in proportion to a movement of the selected object.

18. The apparatus as recited in Claim 17, wherein the object position includes a preferred contact area.

19. The apparatus as recited in Claim 18, wherein the preferred contact area includes a definable point associated with an object that can be selectively moved within the graphical user interface.

20. The apparatus as recited in Claim 17, further comprising an input device operatively coupled to the logic and configured to generate updated positioning information included within the input position.

21. The apparatus as recited in Claim 20, wherein the logic further implements a corrective function that selectively and incrementally reduces the offset based on the updated positioning information.

22. The apparatus as recited in Claim 21, wherein the corrective function includes a linear corrective factor.

23. An apparatus comprising:

a display device having a plurality of pixels;

an input device configured to generate updated positioning information within an input position;

logic operatively coupled to the display device and the input device and configured to determine an offset value between a selected object's position and the input position, and reduce the offset value using a corrective function that selectively and incrementally reduces the offset in proportion to a movement of the selected object based on the updated positioning information.

24. The apparatus as recited in Claim 20, wherein the input device includes a pointing device.

25. The apparatus as recited in Claim 24, wherein the pointing device includes a mouse.

26. The apparatus as recited in Claim 20, wherein the input device includes a touch screen device.

27. The apparatus as recited in claim 17, wherein the logic is operatively configured within a computer.

APPENDIX: EVIDENCE

None.

APPENDIX: RELATED PROCEEDINGS

None.